F 9

```
LLL                 IIIIIIIII     NNN         NNN  KKK         KKK  EEEEEEEEEEEEEEE  RRRRRRRRRRRR
LLL                 IIIIIIIII     NNN         NNN  KKK         KKK  EEEEEEEEEEEEEEE  RRRRRRRRRRRR
LLL                 IIIIIIIII     NNN         NNN  KKK         KKK  EEEEEEEEEEEEEEE  RRRRRRRRRRRR
LLL                    III        NNN         NNN  KKK         KKK  EEE              RRR         RRR
LLL                    III        NNN         NNN  KKK         KKK  EEE              RRR         RRR
LLL                    III        NNN         NNN  KKK         KKK  EEE              RRR         RRR
LLL                    III        NNNNNN      NNN  KKK      KKK     EEE              RRR         RRR
LLL                    III        NNNNNN      NNN  KKK      KKK     EEE              RRR         RRR
LLL                    III        NNNNNN      NNN  KKK      KKK     EEE              RRR         RRR
LLL                    III        NNN   NNN   NNN  KKKKKKKKK        EEEEEEEEEEEE     RRRRRRRRRRRR
LLL                    III        NNN   NNN   NNN  KKKKKKKKK        EEEEEEEEEEEE     RRRRRRRRRRRR
LLL                    III        NNN   NNN   NNN  KKKKKKKKK        EEEEEEEEEEEE     RRRRRRRRRRRR
LLL                    III        NNN      NNNNNN  KKK      KKK     EEE              RRR  RRR
LLL                    III        NNN      NNNNNN  KKK      KKK     EEE              RRR  RRR
LLL                    III        NNN      NNNNNN  KKK      KKK     EEE              RRR  RRR
LLL                    III        NNN         NNN  KKK         KKK  EEE              RRR      RRR
LLL                    III        NNN         NNN  KKK         KKK  EEE              RRR      RRR
LLL                    III        NNN         NNN  KKK         KKK  EEE              RRR      RRR
LLLLLLLLLLLLLLL     IIIIIIIII     NNN         NNN  KKK         KKK  EEEEEEEEEEEEEEE  RRR         RRR
LLLLLLLLLLLLLLL     IIIIIIIII     NNN         NNN  KKK         KKK  EEEEEEEEEEEEEEE  RRR         RRR
LLLLLLLLLLLLLLL     IIIIIIIII     NNN         NNN  KKK         KKK  EEEEEEEEEEEEEEE  RRR         RRR
```

```
**FILE**ID**LNKPROLIB                                  B  6




LL          NN      NN  KK      KK  PPPPPPPP    RRRRRRRR      000000    LL            IIIIII    BBBBBBBB
LL          NN      NN  KK      KK  PPPPPPPP    RRRRRRRR      000000    LL            IIIIII    BBBBBBBB
LL          NN      NN  KK      KK  PP      PP  RR      RR  00      00  LL              II      BB      BB
LL          NN      NN  KK      KK  PP      PP  RR      RR  00      00  LL              II      BB      BB
LL          NNNN    NN  KK    KK    PP      PP  RR      RR  00      00  LL              II      BB      BB
LL          NNNN    NN  KK    KK    PP      PP  RR      RR  00      00  LL              II      BB      BB
LL          NN  NN  NN  KKKKKK      PPPPPPPP    RRRRRRRR    00      00  LL              II      BBBBBBBB
LL          NN  NN  NN  KKKKKK      PPPPPPPP    RRRRRRRR    00      00  LL              II      BBBBBBBB
LL          NN    NNNN  KK    KK    PP          RR  RR      00      00  LL              II      BB      BB
LL          NN    NNNN  KK    KK    PP          RR  RR      00      00  LL              II      BB      BB
LL          NN      NN  KK      KK  PP          RR    RR    00      00  LL              II      BB      BB      ....
LL          NN      NN  KK      KK  PP          RR    RR    00      00  LL              II      BB      BB      ....
LLLLLLLLLL  NN      NN  KK      KK  PP          RR      RR    000000    LLLLLLLLLL    IIIIII    BBBBBBBB        ....
LLLLLLLLLL  NN      NN  KK      KK  PP          RR      RR    000000    LLLLLLLLLL    IIIIII    BBBBBBBB        ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
module lnk_procslib (                                   ! OBJECT LIBRARY PROCESSING
                ident = 'V04-000',
                addressing_mode (external = general, nonexternal = long_relative)
                ) =
begin
!
!****************************************************************************
!*                                                                          *
!*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
!*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
!*  ALL RIGHTS RESERVED.                                                    *
!*                                                                          *
!*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
!*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
!*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
!*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
!*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
!*  TRANSFERRED.                                                            *
!*                                                                          *
!*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
!*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
!*  CORPORATION.                                                            *
!*                                                                          *
!*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
!*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
!*                                                                          *
!*                                                                          *
!****************************************************************************
!
!++
! FACILITY:     LINKER
!
! ABSTRACT:     ROUTINES TO DO ALL PASS 1 OBJECT LIBRARY PROCESSING
!
!
! ENVIRONMENT:  VMS NATIVE MODE
!
! AUTHOR:       T.J. PORTER, CREATION DATE: 16-MAY-77
!
! MODIFIED BY:
!
!       V03-011 JWT0168         Jim Teague              21-Mar-1984
!               LBR$SEARCH will now return a status other than true,
!               so when the Linker returns a 0 from LNK$ADDIMAGE to
!               stop the library search, it must be prepared to see
!               that 0 propagated all the way back through the
!               LBR$SEARCH call.
!
!       V03-010 JWT0099         Jim Teague              14-Mar-1983
!               New CLI interface.
!
!       V03-009 JWT0063         Jim Teague              26-Oct-1982
!               Correct bug in shareable image name manipulation.
!
!       V03-008 JWT0044         Jim Teague              30-Jul-1982
!               Open file performance boost.  Also correct weak
!               shr-img-symbol bug.
```

```
!
!       V03-007 BLS0170         Benn Schreiber          13-Apr-1982
!               Beef up error handling from lbr$ calls
!
!       V03-006 BLS0159         Benn Schreiber          17-Mar-1982
!               Also check for angles in directory spec
!
!--
!
! INCLUDE FILES:
!

library 'STARLETL32';                                   ! STARLET DATA STRUCTURES

require 'PREFIX';                                       ! GENERAL DEFINITIONS

library 'DATBAS';                                       ! INTERNAL DATA BASE

forward routine
    lnk$bintim,                                         ! CONVERT TIME TO BINARY
    lnk$addimage;                                       ! ADD SHAREABLE IMAGE TO CLUSTER LIST

!
! EQUATED SYMBOLS:
!
global literal
    lnk$k_libblocks = 10 : short;                       ! NUMBER OF BLOCKS IN A WINDOW
                                                        ! OF A LIBRARY
!
! EXTERNAL REFERENCES:
!
external literal
    lbr$_keynotfnd,                                     ! KEY NOT FOUND
    lin$_format,                                        ! FORMAT BAD
    lin$_libfind,                                       ! FIND FAILURE IN LIBRARY
    lin$_libnamlng,                                     ! ILLEGAL MODULE NAME LENGTH
    lin$_nosuchmod,                                     ! MODULE NOT IN LIBRARY ERROR
    lin$_readerr;                                       ! READ ERROR

external
    lbr$gl_rmsstv,                                      ! STV RETURNED BY LIBRARIAN
    lnk$gl_ctlmsk : block [, byte],                     ! LINKER CONTROL FLAGS
    lnk$gl_curfil : ref block [, byte],                 ! POINTER TO CURRENT (LIBRARY) FILE DESCRIPTOR
    lnk$gl_curclu : ref block [, byte],                 ! POINTER TO CURRENT CLUSTER DESCRIPTOR
    lnk$gl_clulst,                                      ! HEAD OF CLUSTER DESCRIPTOR LIST
    lnk$gl_clutree,                                     ! TREE HEAD OF CLUSTER TREE
    lnk$gl_lastclu : ref block [, byte],                ! POINTER TO LAST CLUSTER DESCRIPTOR
    lnk$gl_udflst,                                      ! UNDEFINED SYMBOL LISTHEAD
    lnk$gw_nudfsyms : word,                             ! NUMBER OF UNDEFINED SYMBOLS
    lnk$gl_objrecs,                                     ! NUMBER OF RECORDS PROCESSED
    lnk$gb_pass : byte,                                 ! LINKER PASS
    lnk$al_rab : block [rab$c_bln, byte];               ! RAB TO USE FOR READS

external routine
    lib$lookup_tree,                                    ! LOOKUP ITEM IN TREE
    lbr$find,                                           ! POINT TO MODULE
    lbr$set_module,                                     ! READ MODULE HEADER
```

```
1       lbr$get_record,                                 ! READ RECORD OF MODULE
1       lbr$lookup_key,                                 ! LOOKUP KEY IN LIBRARY
1       lbr$set_index,                                  ! SET INDEX NUMBER
1       lbr$search,                                     ! SEARCH INDEX FOR ENTRIES
1       lnk$alloblk,                                    ! DYNAMIC MEMORY ALLOCATOR
1       lnk$dealblk,                                    ! AND DEALLOCATOR
1       lnk$allocluster,                                ! ALLOCATE CLUSTER DESCRIPTOR
1       lnk$insert_clu,                                 ! INSERT CLUSTER INTO CLUSTER TREE
1       lnk$allofdb,                                    ! ALLOCATE FILE DESCRIPTOR BLOCK
1       lnk$procsobj,                                   ! PROCESS OBJ FILES
1       lnk$pointobj;                                   ! POINT TO OBJ IN A LIBRARY
1   !
1   ! MODULE OWN STORAGE:
1   !
1   own
1       shrdefext : quadvector [1] initial (stringdesc ('SYS$LIBRARY:.EXE')),
1                                                       ! DEFAULT NAME STRING FOR SHR IMAGES
1       savedrecount,                                   ! RECORD COUNT A BEGINNING OF LIBRARY MODULE
1       modnamindex : initial (1),                      ! MODULE NAME INDEX IS INDEX 1
1       gstnamindex : initial (2),                      ! GLOBAL SYMBOL INDEX IS INDEX 2
1       gstmisscnt;                                     ! NUMBER OF UNSUCCESSFUL GST SEARCHS THIS CALL
1
1   global
1       lnk$gl_futlsrch,                                ! ACCUMULATED FUTILE SEARCHES
1       lnk$gl_librecs,                                 ! NUMBER OF RECORDS PROCESSED IN LIBRARIES
1       lnk$gl_libsym : ref block [, byte],             ! POINTER TO THE SYMBOL THAT CAUSED
1       lnk$gl_nmodsexp,                                ! NUMBER OF EXPLICITLY EXTRACTED MODULES
1       lnk$gl_nmodsrch;                                !   ''     EXTRACTED BECAUSE THEY RESOLVE SYMBOLS
1                                                       ! A MODULE TO LOAD FROM LIBRARY.
1   literal
1       lnk$k_stopsearch = 0;                           ! Flag to stop library search
1
```

```
1 global routine lnk$procslib (arglist) =         ! PROCESS LIBRARY
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 ! THIS ROUTINE IS CALLED DURING PASS 1 OF
1 ! LINKING TO PROCESS A RELOCATABLE OBJECT MODULE LIBRARY
1 ! WHICH HAS ALREADY BEEN OPENED. THERE ARE TWO FUNCTIONS
1 ! PERFORMED, (IN ORDER IF BOTH SPECIFIED):
1 !         (1) IF EXPLICIT MODULE INCLUSION HAS BEEN SPECIFIED,
1 !             THE NAMED MODULES ARE SEARCHED FOR IN THE
1 !             LIBRARY'S MODULE NAME TABLE AND, IF  FOUND,
1 !             PROCESSED SEQUENTIALLY BY CALLING LNK$PROCSOBJ FOR EACH.
1 !         (2) IF SEARCH FOR UNRESOLVED SYMBOLS IS SPECIFIED, AND THERE
1 !             EXIST CURRENTLY UNDEFINED SYMBOLS ON THE UNDEFINED LIST,
1 !             SEARCH THE LIBRARY GLOBAL SYMBOL TABLE FOR EACH SYMBOL.
1 !             WHEN ONE IS FOUND, PROCESS THE DEFINING MODULE BY
1 !             CALLING LNK$PROCSOBJ.
1 !
1 ! FORMAL PARAMETERS:
1 !         ARGLIST IS THE ADDRESS OF THE ORIGINAL ARGUMENT LIST FROM
1 !         THE IMAGE ACTIVATOR. AT OFFSET CLI$A_UTILSERV IS THE
1 !         ADDRESS AT WHICH TO RE-CALL CLI TO PROVIDE THE MODULE
1 !         NAMES ON AN EXPLICIT MODULE EXTRACTION FROM LIBRARY
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !         LNK$GL_CURFIL - POINTS TO CURRENT OBJ FILE (IN THIS
1 !                         CASE A LIBRARY) DESCRIPTOR BLOCK.
1 !         FLAG BITS IN THE DESCRIPTOR SPECIFY THE KIND OF
1 !         LIBRARY SEARCH (MODULE OR SYMBOL OR BOTH). IF
1 !         MODULE SEARCH IS SPECIFIED, THE FILE DESCRIPTOR CONTAINS
1 !         THE POINTERS TO THE CLI DATA WHICH DESCRIBES MODULES TO
1 !         BE INCLUDED.
1 !         LNK$GW_NUDFSYMS - NUMBER OF UNDEFINED (STRONGLY REFERENCED)
1 !                           SYMBOLS
1 !         LNK$GL_UDFLST -   LISTHEAD FOR DOUBLY LINKED LIST OF
1 !                           UNDEFINED SYMBOLS.
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !         THE MODULES SELECTED FOR PROCESSING ARE PROCESSED BY LNK$PROCSOBJ
1 !         IN ADDITION:
1 !                 LNK$GL_LIBSYM   RECEIVES THE ADDRESS OF THE
1 !                                 ENTRY IN THE SYMBOL TABLE WHEN
1 !                                 A SYMBOL SEARCH IS SUCCESSFUL.
1 !
1 ! ROUTINE VALUE:
1 !
1 ! COMPLETION CODES:
1 !
1 !         NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !         AS PERFORMED BY LNK$PROCSOBJ
1 !
1 !--
```

```
2       begin
2       local
2           moduleptr,
2           nextptr,
2           status,
2           keydesc : block [dsc$c_s_bln, byte],       ! STRING DESCRIPTOR
2           nextsym : ref block [, byte],              ! NEXT UNDEFINED SYMBOL IN LIST
2           modulerfa : block [6, byte];               ! FILE ADDRESS OF FIRST RECORD OF
2                                                      ! THE ASSOCIATED MODULE. FIRST 4
2                                                      ! BYTES ARE VBN, FOLLOWED BY THE
2                                                      ! OFFSET INTO BLOCK
2       map
2           arglist : ref block [, byte];
2       bind
2           auxfnb = lnk$gl_curfil [fdb$t_auxfnb] : block [nam$c_bln, byte];
2                                                      ! AUXILLIARY FILE NAME BLOCK IN FDB
2       if not .lnk$gl_curfil [fdb$v_libextr]          ! IF NOT EXTRACTING SPECIFIC MODULES AND
2       then
2           if .lnk$gw_nudfsyms eql 0 then return true;    ! THERE ARE NO UNDEFINED SYMBOLS
2
2       if .lnk$gl_curfil [fdb$v_libextr]              ! IF THIS IS INCLUSION OF EXPLICITLY
2       then
3           begin
3           lnk$gl_libsym = 0;                         ! INVALIDATE LIBRARY SYMBOL
3           status = lbr$set_index (%ref (.lnk$gl_curfil [fdb$w_ifi]), modnamindex);
3                                                      ! SET TO LOOK AT MODULE NAME INDEX
3           moduleptr = .lnk$gl_curfil[fdb$l_omdlst];
3           lnk$gl_curfil[fdb$l_omdlst] = 0;
3
3           if not .status
3           then
4               begin
4               signal (lin$_readerr, 1, lnk$gl_curfil [fdb$q_filename], .status);
4               return true;                           ! DON'T ABORT THE LINK, THO
3               end;
3   !
3   ! NOW LOOP, GRABBING THE NEXT MODULE NAME IN THE LINKED
3   ! LIST, SEARCHING MODULE NAME TABLE FOR THAT MODULE THEN,
3   ! IF FOUND PROCESSING THE MODULE
3   !
3           while .moduleptr neq 0                     ! THAT IS WHILE THERE
3           do
4               begin                                  ! REMAINS MORE TEXT ON THE
4               nextptr = .(.moduleptr);
4               keydesc[dsc$w_length] = .(.moduleptr+4)<0,8>;
4               keydesc[dsc$a_pointer] = .moduleptr + 5;
4               if .keydesc [dsc$w_length] eql 0    ! GO GET NEXT NAME (ALLOWING
4                   or .keydesc [dsc$w_length] gtru sym$c_maxlng
4                                                      ! CLI TO USE THE LIBRARY HEADER BUFFER)
4               then
5                   begin
5                   signal (lin$_libnamlng, 2,        ! CHECK A VALID NAME
5                       keydesc [dsc$w_length], .keydesc[dsc$w_length]);    ! AND ISSUE ERROR IF AN
5                   keydesc [dsc$w_length] = sym$c_maxlng;  ! ILLEGAL LENGTH, SET TO MAXIMUM
4                   end;
5               if not (status = lbr$lookup_key (%ref (.lnk$gl_curfil [fdb$w_ifi]), keydesc, modulerfa))
5                                                      ! LOOKUP THE MODULE NAME
```

```
4                then
5                    begin
5                    if .status eql lbr$_keynotfnd
5                    then
5                        signal (lin$_nosuchmod, 2, keydesc [dsc$w_length],
5                            lnk$gl_curfil [fdb$q_filename])
5                    else
5                        signal (lin$_readerr, 1, lnk$gl_curfil [fdb$q_filename], .status, .lbr$gl_rmsstv);
5                    end
4                else
5                    begin
5                    if .lnk$gl_curfil [fdb$v_imglib]        ! IF THIS IS SHR IMG STB LIBRARY
5                    then
5                        lnk$addimage (keydesc, modulerfa)   !  THEN JUST ADD TO THE CLUSTER LIST
5                    else
6                        begin
6                        savedrecount = .lnk$gl_objrecs;     ! SAVE CURRENT RECORD COUNT
6                        lnk$gl_nmodsexp = .lnk$gl_nmodsexp + 1;     ! COUNT ONE MORE EXPLICITLY EXTRACTED
6                        lnk$pointobj (modulerfa);   ! FOUND IT SO GO POINT TO
6
6                        if not lnk$procsobj (modulerfa) then return false;  ! THE MODULE IN THE LIBRARY
6                        lnk$gl_librecs = .lnk$gl_librecs + .lnk$gl_objrecs -
6                                                            ! ACCUMULATE THE NUMBER OF RECORDS
6                        .savedrecount;                      ! FOUND IN LIBRARIES
5                        end;
4                    end;
4                lnk$dealblk(.keydesc[dsc$w_length]+5, .moduleptr);
4                moduleptr = .nextptr;
3                end;
2        end;                                                ! AND PROCESS IT
2 !
2 ! NOW CHECK WHETHER THIS LIBRARY IS TO BE SEARCHED FOR
2 ! CURRENTLY UNDEFINED SYMBOLS. EXIT NOW IF NOT
2 !
2    if .lnk$gl_curfil [fdb$v_libsrch]               ! IF A SYMBOL SEARCH REQUIRED
2    then
3        begin
3        lnk$gl_curfil [fdb$v_newudf] = false;       ! RESET UNDEFINED SYMBOLS CONTRIBUTED
3        gstmisscnt = 0;                             ! RESET COUNT OF SYMBOLS NOT FOUND
3        nextsym = .lnk$gl_udflst;                   ! START AT TOP OF LIST, AND
3        status = lbr$set_index (%ref (.lnk$gl_curfil [fdb$w_ifi]), gstnamindex);
3                                                    ! LOOK IN GLOBAL SYMBOL INDEX
3        if not .status
3        then
4            begin
4            signal (lin$_readerr, 1, lnk$gl_curfil [fdb$q_filename], .status);
4            return true;                            ! DON'T ABORT THE LINK, THO
3            end;
3
3        if .lnk$gl_curfil [fdb$v_imglib]            ! IF THIS IS SHR IMG STB LIBRARY
3        then
4            begin
4            while .nextsym neq lnk$gl_udflst do
5                begin
5                bind
5                    nextsymnam = .nextsym - .nextsym [sym$b_namlng] - snb$c_fxdlen : block [, byte];
5                if not .nextsym [sym$v_weak]
```

```
5          then
6              begin
6              keydesc [dsc$w_length] = .nextsym [sym$b_namlng];
6              keydesc [dsc$a_pointer] = nextsymnam [snb$t_name];
6
7              if (status = lbr$lookup_key (%ref (.lnk$gl_curfil [fdb$w_ifi]), keydesc, modulerfa))
7                                           ! IF SYMBOL IS IN LIBRARY
6              then
7                  begin
7                  status = lbr$search (%ref (.lnk$gl_curfil [fdb$w_ifi]), modnamindex,
7                                           ! FIND THE MODULE NAME
7                      modulerfa, lnk$addimage);
8                  if (not .status)  and (.status neq lnk$k_stopsearch)
7                  then
7                      signal (lin$_readerr, 1, lnk$gl_curfil [fdb$q_filename], .status);
7                  end
6              else
6
6                  if .status neq lbr$_keynotfnd
6                  then
6                      signal (lin$_readerr, 1,
6                          lnk$gl_curfil [fdb$q_filename], .status, .lbr$gl_rmsstv);
5              end;
5
5          nextsym = .nextsym [sym$l_udflink];     ! LINK TO NEXT UNDEFINED SYMBOL
5          end
4      end
3  else
3
3      while .lnk$gw_nudfsyms neq 0            ! WHILE IT CONTAINS SOME UN-
3                                              ! DEFINED SYMBOLS, GET
4          and (if (lnk$gl_libsym = .nextsym) neq lnk$gl_udflst    ! NEXT ENTRY. HOWEVER
4          then true                           ! IF BACK AT THE LISTHEAD
4          else if not .lnk$gl_curfil [fdb$v_newudf]       ! AND THIS FILE DID NOT ADD
4              then false                      ! MORE UNDEFINED SYMBOLS-WE ARE DONE
4              else
5                  begin                       ! IF IT DID ADD MORE, GET
5                  lnk$gl_libsym = .lnk$gl_libsym [sym$l_udflink]; ! TOP ENTRY IN LIST
5                  lnk$gl_curfil [fdb$v_newudf] = false;   ! RESET THE UNDEFINED SYMBOLS CONTRIBUTED FL
5                  true                        ! AND CONTINUE THE
5                  end
4          )                                   ! SEARCH
3      do
4          begin                               ! FOR A SYMBOL ON THE
4
4          bind
4              libsymnam = .lnk$gl_libsym - .lnk$gl_libsym [sym$b_namlng] - snb$c_fxdlen : block [, byt
4          ;                                   ! POINT TO NAME PART
4
4          keydesc [dsc$w_length] = .lnk$gl_libsym [sym$b_namlng]; ! MAKE STRING DESCRIPTOR FOR NAME
4          keydesc [dsc$a_pointer] = libsymnam [snb$t_name];        ! ...
4          nextsym = .lnk$gl_libsym [sym$l_udflink];        ! UNDEFINED LIST AND
4
4          if (.lnk$gl_libsym [sym$w_flags] and gsy$m_weak) eql 0  ! PROVIDED IT IS NOT A WEAK
4              and not .lnk$gl_libsym [sym$v_gstmiss]      ! REFERENCE AND THAT WE
4          then                                ! HAVE NOT BEFORE FAILED TO
4                                              ! FIND IT IN THIS LIBRARY,
;  376        0489  4
;  377        0490  5                     if (status = lbr$lookup_key (%ref (.lnk$gl_curfil [fdb$w_ifi]), keydesc, modulerfa))
;  378        0491  5                                                    ! GO LOOK FOR THE SYMBOL
;  379        0492  4                     then
;  380        0493  5                         begin                      ! RETURN RECORD'S FILE ADDRESS
;  381        0494  5                         lnk$gl_nmodsrch = .lnk$gl_nmodsrch + 1; ! COUNT THE NUMBER OF MODULES
;  382        0495  5                         savedrecount = .lnk$gl_objrecs; ! SAVE CURRENT RECORD COUNT
;  383        0496  5                         lnk$pointobj (modulerfa);       ! TO POINT TO THE MODULE
;  384        0497  5
;  385        0498  5                         if not lnk$procsobj (modulerfa) then return false;      ! AND GO PROCESS IT
;  386        0499  5                         lnk$gl_librecs = .lnk$gl_librecs + .lnk$gl_objrecs -    ! ACCUMULATE THE NUMBER OF
;  387        0500  5                         .savedrecount;             ! RECORDS FROM LIBRARIES
;  388        0501  5                         nextsym = .lnk$gl_libsym;         ! RETRIEVE NEXT IN LIST (SINCE THE
;  389        0502  5                                                    ! ONE WE HAD MAY HAVE JUST BEEN
;  390        0503  5                                                    ! DEFINED BY THAT MODULE)
;  391        0504  5                         end
;  392        0505  4                     else
;  393        0506  5                         begin                      ! IF THE SYMBOL WAS NOT
;  394        0507  5
;  395        0508  5                         if .status neq lbr$_keynotfnd
;  396        0509  5                         then
;  397        0510  5                             signal (lin$_readerr, 1,
;  398        0511  5                                 lnk$gl_curfil [fdb$q_filename], .status, .lbr$gl_rmsstv);
;  399        0512  5
;  400        0513  5                         gstmisscnt = .gstmisscnt + 1;    ! FOUND IN LIBRARY, COUNT
;  401        0514  5                         lnk$gl_libsym [sym$v_gstmiss] = true;    ! ANOTHER MISS AND SUPPRESS
;  402        0515  5                                                    ! ANY MORE SEARCHES FOR IT
;  403        0516  4                         end;                       ! END OF SYMBOL LIST LOOP
;  404        0517  3                 end;
;  405        0518  3         lnk$gl_libsym = 0;                         ! INVALIDATE THE SYMBOL POINTER
;  406        0519  3 !
;  407        0520  3 ! NOW FINISHED LOOKING FOR UNDEFINED SYMBOLS IN THE CURRENT LIBRARY
;  408        0521  3 ! MUST NOW GO DOWN WHAT IS LEFT OF THE UNDEFINED SYMBOL LIST, TURNING
;  409        0522  3 ! OFF THE GST MISS FLAG IN EACH SYMBOL DESCRIPTOR.
;  410        0523  3 !
;  411        0524  3         nextsym = lnk$gl_udflst;
;  412        0525  3         if .gstmisscnt neq 0                       ! IF THERE WERE NO MISSES
;  413        0526  3         then
;  414        0527  3             while (nextsym = .nextsym [sym$l_udflink]) neq lnk$gl_udflst        ! FORGET IT
;  415        0528  3             do
;  416        0529  3                 nextsym [sym$v_gstmiss] = false;        ! TURN OFF FLAG
;  417        0530  3
;  418        0531  3         lnk$gl_futlsrch = .lnk$gl_futlsrch + .gstmisscnt;       ! ACCUMULATE FUTILE SEARCH COUNT
;  419        0532  2         end;
;  420        0533  2
;  421        0534  2     lnk$gl_curfil [fdb$v_selser] = false;          ! RESET THE POSSIBLE SELECTIVE SEARCH FLAG
;  422        0535  2     return true;                                   ! AND ALL DONE
;  423        0536  1     end;                                           ! END OF ROUTINE


                                                        .TITLE  LNK_PROCSLIB
                                                        .IDENT  \V04-000\

                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

58 45 2E 3A 59 52 41 52 42 49 4C 24 53 59 53 00000 P.AAA:  .ASCII  \SYS$LIBRARY:.EXE\                        ;
                                          45 0000F                                                  ;
```

```
                                        .PSECT  $OWN$,NOEXE,2

                 00000010  00000 SHRDEFEXT:
                                        .LONG   16                                        ;
                 00000000' 00004        .ADDRESS P.AAA                                    ;
                           00008 SAVEDRECOUNT:
                                        .BLKB   4
                 00000001  0000C MODNAMINDEX:
                                        .LONG   1                                         ;
                 00000002  00010 GSTNAMINDEX:
                                        .LONG   2                                         ;
                           00014 GSTMISSCNT:
                                        .BLKB   4

                                        .PSECT  $GLOBAL$,NOEXE,2

                           00000 LNK$GL_FUTLSRCH::
                                        .BLKB   4
                           00004 LNK$GL_LIBRECS::
                                        .BLKB   4
                           00008 LNK$GL_LIBSYM::
                                        .BLKB   4
                           0000C LNK$GL_NMODSEXP::
                                        .BLKB   4
                           00010 LNK$GL_NMODSRCH::
                                        .BLKB   4

                                 LNK$K_LIBBLOCKS==    10
                                        .EXTRN  LBR$_KEYNOTFND, LIN$_FORMAT
                                        .EXTRN  LIN$_LIBFIND, LIN$_LIBNAMLNG
                                        .EXTRN  LIN$_NOSUCHMOD, LIN$_READERR
                                        .EXTRN  LBR$GL_RMSSTV, LNK$GL_CTLMSK
                                        .EXTRN  LNK$GL_CURFIL, LNK$GL_CURCLU
                                        .EXTRN  LNK$GL_CLULST, LNK$GL_CLUTREE
                                        .EXTRN  LNK$GL_LASTCLU, LNK$GL_UDFLST
                                        .EXTRN  LNK$GW_NUDFSYMS
                                        .EXTRN  LNK$GL_OBJRECS, LNK$GB_PASS
                                        .EXTRN  LNK$AL_RAB, LIB$LOOKUP_TREE
                                        .EXTRN  LBR$FIND, LBR$SET_MODULE
                                        .EXTRN  LBR$GET_RECORD, LBR$LOOKUP_KEY
                                        .EXTRN  LBR$SET_INDEX, LBR$SEARCH
                                        .EXTRN  LNK$ALLOBLK, LNK$DEALBLK
                                        .EXTRN  LNK$ALLOCLUSTER
                                        .EXTRN  LNK$INSERT_CLU, LNK$ALLOFDB
                                        .EXTRN  LNK$PROCSOBJ, LNK$POINTOBJ

                                        .PSECT  $CODE$,NOWRT,2

                      0FFC 00000        .ENTRY  LNK$PROCSLIB, Save R2,R3,R4,R5,R6,R7,R8,R9,-; 0261
                                                R10,R11                                   ;
5B 00000000G   00   9E 00002        MOVAB   LNK$GL_OBJRECS, R11                       ;
5A 00000000G   00   9E 00009        MOVAB   LNK$GL_UDFLST, R10                        ;
59 00000000G   8F   D0 00010        MOVL    #LIN$_READERR, R9                         ;
58 00000000G   00   9E 00017        MOVAB   LIB$SIGNAL, R8                            ;
57 00000000'   EF   9E 0001E        MOVAB   SAVEDRECOUNT, R7                          ;
56 00000000'   EF   9E 00025        MOVAB   LNK$GL_LIBSYM, R6                         ;
```

```
LNK_PROCSLIB                                      L 6
V04-000                             16-Sep-1984 00:21:45   VAX-11 Bliss-32 V4.0-742        Page 10
                                    14-Sep-1984 12:40:34   [LINKER.SRC]LNKPROLIB.B32;1      (2)

                 55  00000000G   00  9E  0002C        MOVAB   LNK$GL_CURFIL, R5                 :
                 5E              14  C2  00033        SUBL2   #20, SP                           :
                 50              65  D0  00036        MOVL    LNK$GL_CURFIL, R0                 : 0332
13         0A    A0              06  E0  00039        BBS     #6, 10(R0), 3$                    : 0334
                     00000000G   00  B5  0003E        TSTW    LNK$GW_NUDFSYMS                   : 0336
                                 03  12  00044        BNEQ    1$                                :
                               02C1  31  00046        BRW     31$                               :
03         0A    A0              06  E0  00049  1$:   BBS     #6, 10(R0), 3$                    : 0338
                               0100  31  0004E  2$:   BRW     12$                               :
                                 66  D4  00051  3$:   CLRL    LNK$GL_LIBSYM                     : 0341
                          04     A7  9F  00053        PUSHAB  MODNAMINDEX                       : 0342
           04    AE       24     A0  3C  00056        MOVZWL  36(R0), 4(SP)                     :
                          04     AE  9F  0005B        PUSHAB  4(SP)                             :
       00000000G 00              02  FB  0005E        CALLS   #2, LBR$SET_INDEX                 :
                 53              50  D0  00065        MOVL    R0, STATUS                        :
                 50              65  D0  00068        MOVL    LNK$GL_CURFIL, R0                 : 0344
                 52       04     A0  D0  0006B        MOVL    4(R0), MODULEPTR                  :
                          04     A0  D4  0006F        CLRL    4(R0)                             : 0345
                 08              53  E8  00072        BLBS    STATUS, 4$                        : 0347
                                 53  DD  00075        PUSHL   STATUS                            : 0350
                          14     A0  9F  00077        PUSHAB  20(R0)                            :
                               0107  31  0007A        BRW     14$                               :
                                 52  D5  0007D  4$:   TSTL    MODULEPTR                         : 0358
                                 CD  13  0007F        BEQL    2$                                :
                 54              62  D0  00081        MOVL    (MODULEPTR), NEXTPTR              : 0361
           0C    AE       04     A2  9B  00084        MOVZBW  4(MODULEPTR), KEYDESC             : 0362
           10    AE       05     A2  9E  00089        MOVAB   5(R2), KEYDESC+4                  : 0363
                 50       0C     AE  3C  0008E        MOVZWL  KEYDESC, R0                       : 0364
                                 05  13  00092        BEQL    5$                                :
                 1F              50  B1  00094        CMPW    R0, #31                           : 0365
                                 14  1B  00097        BLEQU   6$                                :
                                 50  DD  00099  5$:   PUSHL   R0                                : 0370
                          10     AE  9F  0009B        PUSHAB  KEYDESC                           :
                                 02  DD  0009E        PUSHL   #2                                :
                     00000000G   8F  DD  000A0        PUSHL   #LIN$_LIBNAMLNG                   :
                 68              04  FB  000A6        CALLS   #4, LIB$SIGNAL                    :
           0C    AE              1F  B0  000A9        MOVW    #31, KEYDESC                      : 0371
                          04     AE  9F  000AD  6$:   PUSHAB  MODULERFA                         : 0373
                          10     AE  9F  000B0        PUSHAB  KEYDESC                           :
                 50              65  D0  000B3        MOVL    LNK$GL_CURFIL, R0                 :
           08    AE       24     A0  3C  000B6        MOVZWL  36(R0), 8(SP)                     :
                          08     AE  9F  000BB        PUSHAB  8(SP)                             :
       00000000G 00              03  FB  000BE        CALLS   #3, LBR$LOOKUP_KEY                :
                 53              50  D0  000C5        MOVL    R0, STATUS                        :
                 30              53  E8  000C8        BLBS    STATUS, 8$                        :
50               65              14  C1  000CB        ADDL3   #20, LNK$GL_CURFIL, R0            : 0380
       00000000G 8F              53  D1  000CF        CMPL    STATUS, #LBR$_KEYNOTFND           : 0377
                                 12  12  000D6        BNEQ    7$                                :
                                 50  DD  000D8        PUSHL   R0                                : 0380
                          10     AE  9F  000DA        PUSHAB  KEYDESC                           : 0379
                                 02  DD  000DD        PUSHL   #2                                : 0380
                     00000000G   8F  DD  000DF        PUSHL   #LIN$_NOSUCHMOD                   :
                 68              04  FB  000E5        CALLS   #4, LIB$SIGNAL                    :
                                 51  11  000E8        BRB     11$                               :
                     00000000G   00  DD  000EA  7$:   PUSHL   LBR$GL_RMSSTV                     : 0382
                                 09  BB  000F0        PUSHR   #^M<R0,R3>                        :
                                 01  DD  000F2        PUSHL   #1                                :
                                   59  DD 000F4         PUSHL   R9                                      ;
                     68            05  FB 000F6         CALLS   #5, LIB$SIGNAL                          ;
                                   40  11 000F9         BRB     11$                                     ; 0373
                     50            65  D0 000FB 8$:     MOVL    LNK$GL_CURFIL, R0                       ; 0386
                     0F      0B    A0  E9 000FE         BLBC    11(R0), 9$                              ;
                             04    AE  9F 00102         PUSHAB  MODULERFA                               ; 0388
                             10    AE  9F 00105         PUSHAB  KEYDESC                                 ;
            00000000V  EF          02  FB 00108         CALLS   #2, LNK$ADDIMAGE                        ;
                                   2A  11 0010F         BRB     11$                                     ;
                     67            6B  D0 00111 9$:     MOVL    LNK$GL_OBJRECS, SAVEDRECOUNT            ; 0391
                             04    A6  D6 00114         INCL    LNK$GL_NMODSEXP                         ; 0392
                             04    AE  9F 00117         PUSHAB  MODULERFA                               ; 0393
            00000000G  00          01  FB 0011A         CALLS   #1, LNK$POINTOBJ                        ;
                             04    AE  9F 00121         PUSHAB  MODULERFA                               ; 0395
            00000000G  00          01  FB 00124         CALLS   #1, LNK$PROCSOBJ                        ;
                       03          50  E8 0012B         BLBS    R0, 10$                                 ;
                                 01DD  31 0012E         BRW     32$                                     ;
     50            FC  A6          6B  C1 00131 10$:    ADDL3   LNK$GL_OBJRECS, LNK$GL_LIBRECS, R0      ; 0396
FC   A6                50          67  C3 00136         SUBL3   SAVEDRECOUNT, R0, LNK$GL_LIBRECS        ; 0398
                                   52  DD 0013B 11$:    PUSHL   MODULEPTR                               ; 0401
                       7E    10    AE  3C 0013D         MOVZWL  KEYDESC, -(SP)                          ;
                       6E          05  C0 00141         ADDL2   #5, (SP)                                ;
            00000000G  00          02  FB 00144         CALLS   #2, LNK$DEALBLK                         ;
                       52          54  D0 0014B         MOVL    NEXTPTR, MODULEPTR                      ; 0402
                                 FF2C  31 0014E         BRW     4$                                      ; 0358
                       50          65  D0 00151 12$:    MOVL    LNK$GL_CURFIL, R0                       ; 0409
                             0A    A0  95 00154         TSTB    10(R0)                                  ;
                                   03  19 00157         BLSS    13$                                     ;
                                 01A7  31 00159         BRW     30$                                     ;
                   0A  A0          01  8A 0015C 13$:    BICB2   #1, 10(R0)                              ; 0412
                             0C    A7  D4 00160         CLRL    GSTMISSCNT                              ; 0413
                       52          6A  D0 00163         MOVL    LNK$GL_UDFLST, NEXTSYM                  ; 0414
                             08    A7  9F 00166         PUSHAB  GSTNAMINDEX                             ; 0415
                   04  AE    24    A0  3C 00169         MOVZWL  36(R0), 4(SP)                           ;
                             04    AE  9F 0016E         PUSHAB  4(SP)                                   ;
            00000000G  00          02  FB 00171         CALLS   #2, LBR$SET_INDEX                       ;
                       53          50  D0 00178         MOVL    R0, STATUS                              ;
                       10          53  E8 0017B         BLBS    STATUS, 15$                             ; 0417
                                   53  DD 0017E         PUSHL   STATUS                                  ; 0420
     7E                65          14  C1 00180         ADDL3   #20, LNK$GL_CURFIL, -(SP)               ;
                                   01  DD 00184 14$:    PUSHL   #1                                      ;
                                   59  DD 00186         PUSHL   R9                                      ;
                       68          04  FB 00188         CALLS   #4, LIB$SIGNAL                          ;
                                 017C  31 0018B         BRW     31$                                     ; 0421
                       50          65  D0 0018E 15$:    MOVL    LNK$GL_CURFIL, R0                       ; 0424
                       03    0B    A0  E8 00191         BLBS    11(R0), 16$                             ;
                                 0095  31 00195         BRW     20$                                     ;
                       50          6A  9E 00198 16$:    MOVAB   LNK$GL_UDFLST, R0                       ; 0427
                       50          52  D1 0019B         CMPL    NEXTSYM, R0                             ;
                                   03  12 0019E         BNEQ    17$                                     ;
                                 0140  31 001A0         BRW     27$                                     ;
                       50    0F    A2  9A 001A3 17$:    MOVZBL  15(NEXTSYM), R0                         ; 0430
     50                52          50  C3 001A7         SUBL3   R0, NEXTSYM, R0                         ;
                       78    0A    A2  E8 001AB         BLBS    10(NEXTSYM), 19$                        ; 0431
                   0C  AE    0F    A2  9B 001AF         MOVZBW  15(NEXTSYM), KEYDESC                    ; 0434
                   10  AE          60  9E 001B4         MOVAB   (R0), KEYDESC+4                         ; 0435
                             04    AE  9F 001B8         PUSHAB  MODULERFA                               ; 0437
                             10   AE  9F 001BB         PUSHAB   KEYDESC                             :
                   50             65  D0 001BE         MOVL     LNK$GL_CURFIL, R0                   :
           08      AE        24   A0  3C 001C1         MOVZWL   36(R0), 8(SP)                       :
                             08   AE  9F 001C6         PUSHAB   8(SP)                               :
     00000000G     00             03  FB 001C9         CALLS    #3, LBR$LOOKUP_KEY                  :
                   53             50  D0 001D0         MOVL     R0, STATUS                          :
                   35             53  E9 001D3         BLBC     STATUS, 18$                         :
                       00000000V  EF  9F 001D6         PUSHAB   LNK$ADDIMAGE                        : 0441
                             08   AE  9F 001DC         PUSHAB   MODULERFA                           :
                             04   A7  9F 001DF         PUSHAB   MODNAMINDEX                         :
                   50             65  D0 001E2         MOVL     LNK$GL_CURFIL, R0                   :
           0C      AE        24   A0  3C 001E5         MOVZWL   36(R0), 12(SP)                      :
                             0C   AE  9F 001EA         PUSHAB   12(SP)                              :
     00000000G     00             04  FB 001ED         CALLS    #4, LBR$SEARCH                      :
                   53             50  D0 001F4         MOVL     R0, STATUS                          :
                   2D             53  E8 001F7         BLBS     STATUS, 19$                         : 0444
                                  2B  13 001FA         BEQL     19$                                 :
                                  53  DD 001FC         PUSHL    STATUS                              : 0446
7E                 65             14  C1 001FE         ADDL3    #20, LNK$GL_CURFIL, -(SP)           :
                                  01  DD 00202         PUSHL    #1                                  :
                                  59  DD 00204         PUSHL    R9                                  :
                   68             04  FB 00206         CALLS    #4, LIB$SIGNAL                      :
                                  1C  11 00209         BRB      19$                                 : 0437
     00000000G     8F             53  D1 0020B 18$:    CMPL     STATUS, #LBR$_KEYNOTFND             : 0450
                                  13  13 00212         BEQL     19$                                 :
                       00000000G  00  DD 00214         PUSHL    LBR$GL_RMSSTV                       : 0453
                                  53  DD 0021A         PUSHL    STATUS                              :
7E                 65             14  C1 0021C         ADDL3    #20, LNK$GL_CURFIL, -(SP)           :
                                  01  DD 00220         PUSHL    #1                                  :
                                  59  DD 00222         PUSHL    R9                                  :
                   68             05  FB 00224         CALLS    #5, LIB$SIGNAL                      :
                   52             62  D0 00227 19$:    MOVL     (NEXTSYM), NEXTSYM                  : 0456
                                FF6B  31 0022A         BRW      16$                                 : 0426
                       00000000G  00  B5 0022D 20$:    TSTW     LNK$GW_NUDFSYMS                     : 0461
                                  03  12 00233         BNEQ     22$                                 :
                                00AB  31 00235 21$:    BRW      27$                                 :
                   66             52  D0 00238 22$:    MOVL     NEXTSYM, LNK$GL_LIBSYM              : 0463
                   50             6A  9E 0023B         MOVAB    LNK$GL_UDFLST, R0                   :
                   50             52  D1 0023E         CMPL     NEXTSYM, R0                         :
                                  0E  12 00241         BNEQ     23$                                 :
                   50             65  D0 00243         MOVL     LNK$GL_CURFIL, R0                   : 0465
                   EB        0A   A0  E9 00246         BLBC     10(R0), 21$                         :
                   76             96  D0 0024A         MOVL     @LNK$GL_LIBSYM, LNK$GL_LIBSYM       : 0469
           0A      A0             01  8A 0024D         BICB2    #1, 10(R0)                          : 0470
                   50             66  D0 00251 23$:    MOVL     LNK$GL_LIBSYM, R0                   : 0478
                   51        0F   A0  9A 00254         MOVZBL   15(R0), R1                          :
51                 50             51  C3 00258         SUBL3    R1, R0, R1                          :
           0C      AE        0F   A0  9B 0025C         MOVZBW   15(R0), KEYDESC                     : 0481
           10      AE             61  9E 00261         MOVAB    (R1), KEYDESC+4                     : 0482
                   52             60  D0 00265         MOVL     (R0), NEXTSYM                       : 0483
                   C1        0A   A0  E8 00268         BLBS     10(R0), 20$                         : 0485
                   BD        0C   A0  E8 0026C         BLBS     12(R0), 20$                         : 0486
                             04   AE  9F 00270         PUSHAB   MODULERFA                           : 0490
                             10   AE  9F 00273         PUSHAB   KEYDESC                             :
                   50             65  D0 00276         MOVL     LNK$GL_CURFIL, R0                   :
           08      AE        24   A0  3C 00279         MOVZWL   36(R0), 8(SP)                       :
                             08   AE  9F 0027E         PUSHAB   8(SP)                               :
```

```
                  00000000G  00               03  FB 00281          CALLS   #3, LBR$LOOKUP_KEY                              :
                             53               50  D0 00288          MOVL    R0, STATUS                                      :
                             2C               53  E9 0028B          BLBC    STATUS, 24$                                     :
                                      08      A6  D6 0028E          INCL    LNK$GL_NMODSRCH                                 : 0494
                             67               6B  D0 00291          MOVL    LNK$GL_OBJRECS, SAVEDRECOUNT                    : 0495
                                      04      AE  9F 00294          PUSHAB  MODULERFA                                       : 0496
                  00000000G  00               01  FB 00297          CALLS   #1, LNK$POINTOBJ                                :
                                      04      AE  9F 0029E          PUSHAB  MODULERFA                                       : 0498
                  00000000G  00               01  FB 002A1          CALLS   #1, LNK$PROCSOBJ                                :
                             63               50  E9 002A8          BLBC    R0, 32$                                         :
            50        FC     A6               6B  C1 002AB          ADDL3   LNK$GL_OBJRECS, LNK$GL_LIBRECS, R0              : 0499
     FC     A6               50               67  C3 002B0          SUBL3   SAVEDRECOUNT, R0, LNK$GL_LIBRECS                : 0500
                             52               66  D0 002B5          MOVL    LNK$GL_LIBSYM, NEXTSYM                          : 0501
                             26               11  002B8             BRB     26$                                             : 0490
                  00000000G  8F               53  D1 002BA 24$:     CMPL    STATUS, #LBR$_KEYNOTFND                         : 0508
                             13               13  002C1             BEQL    25$                                             :
                                   00000000G  00  DD 002C3          PUSHL   LBR$GL_RMSSTV                                   : 0511
                                              53  DD 002C9          PUSHL   STATUS                                          :
            7E               65               14  C1 002CB          ADDL3   #20, LNK$GL_CURFIL, -(SP)                       :
                                              01  DD 002CF          PUSHL   #1                                              :
                                              59  DD 002D1          PUSHL   R9                                              :
                             68               05  FB 002D3          CALLS   #5, LIB$SIGNAL                                  :
                                      0C      A7  D6 002D6 25$:     INCL    GSTMISSCNT                                      : 0513
                             50               66  D0 002D9          MOVL    LNK$GL_LIBSYM, R0                               : 0514
                       0C    A0               01  88 002DC          BISB2   #1, 12(R0)                                      :
                                            FF4A  31 002E0 26$:     BRW     20$                                             : 0490
                                              66  D4 002E3 27$:     CLRL    LNK$GL_LIBSYM                                   : 0518
                             52               6A  9E 002E5          MOVAB   LNK$GL_UDFLST, NEXTSYM                          : 0524
                             51       0C      A7  D0 002E8          MOVL    GSTMISSCNT, R1                                  : 0525
                                              11  13 002EC          BEQL    29$                                             :
                             52               62  D0 002EE 28$:     MOVL    (NEXTSYM), NEXTSYM                              : 0527
                             50               6A  9E 002F1          MOVAB   LNK$GL_UDFLST, R0                               :
                             50               52  D1 002F4          CMPL    NEXTSYM, R0                                     :
                                              06  13 002F7          BEQL    29$                                             :
                       0C    A2               01  8A 002F9          BICB2   #1, 12(NEXTSYM)                                 : 0529
                                              EF  11 002FD          BRB     28$                                             :
                       F8    A6               51  C0 002FF 29$:     ADDL2   R1, LNK$GL_FUTLSRCH                             : 0531
                             50               65  D0 00303 30$:     MOVL    LNK$GL_CURFIL, R0                               : 0534
                       0A    A0               08  8A 00306          BICB2   #8, 10(R0)                                      :
                             50               01  D0 0030A 31$:     MOVL    #1, R0                                          : 0535
                                                  04 0030D          RET                                                     :
                                              50  D4 0030E 32$:     CLRL    R0                                              : 0536
                                                  04 00310          RET                                                     :

; Routine Size:  785 bytes,    Routine Base:  $CODE$ + 0000


;  424          0537  1
```

```
LNK_PROCSLIB                                        C 7
V04=000                                    16-Sep-1984 00:21:45    VAX-11 Bliss-32 V4.0-742              Page 14
                                           14-Sep-1984 12:40:34    [LINKER.SRC]LNKPROLIB.B32;1               (3)

;  426        0538  1 global routine lnk$bintim (asctim, bintim) =
;  427        0539  2     begin
;  428        0540  2 !
;  429        0541  2 !       THIS ROUTINE CONVERTS A DATE/TIME STRING FROM A MODULE
;  430        0542  2 !       HEADER TO BINARY.
;  431        0543  2 !
;  432        0544  2 ! INPUTS:
;  433        0545  2 !
;  434        0546  2 !       ASCTIM          ADDRESS OF 17-BYTE ASCII DATE/TIME
;  435        0547  2 !       BINTIM          ADDRESS OF QUADWORD TO STORE BINARY TIME
;  436        0548  2 !
;  437        0549  2 !--
;  438        0550  2     local
;  439        0551  2         timedesc : block [dsc$c_s_bln, byte],
;  440        0552  2         timestring : vector [23, byte];
;  441        0553  2
;  442        0554  2     ch$move (17, .asctim, timestring);          ! COPY ASCII STRING FOR DATE/TIME
;  443        0555  2     ch$fill (%c'0', 6, timestring [17]);        ! FILL REST OF STRING WITH 0'S
;  444        0556  2     timestring [17] = %c':';                    ! FIX PUNCTUATION AS REQUIRED
;  445        0557  2     timestring [20] = %c'.';
;  446        0558  2     timedesc [dsc$w_length] = 23;
;  447        0559  2     timedesc [dsc$a_pointer] = timestring;
;  448        0560  2     $bintim (timbuf = timedesc, timadr = .bintim);
;  449        0561  2     return true
;  450        0562  1     end;


                                                           .EXTRN  SYS$BINTIM

                                       003C 00000          .ENTRY  LNK$BINTIM, Save R2,R3,R4,R5             ; 0538
                        5E          20  C2 00002          SUBL2   #32, SP                                  ;
                 6E     04  BC      11  28 00005          MOVC3   #17, @ASCTIM, TIMESTRING                 ; 0554
      06         30         6E      00  2C 0000A          MOVC5   #0, (SP), #48, #6, TIMESTRING+17         ; 0555
                               11   AE     0000F
                        11  AE      3A  90 00011          MOVB    #58, TIMESTRING+17                       ; 0556
                        14  AE      2E  90 00015          MOVB    #46, TIMESTRING+20                       ; 0557
                        18  AE      17  B0 00019          MOVW    #23, TIMEDESC                            ; 0558
                        1C  AE      6E  9E 0001D          MOVAB   TIMESTRING, TIMEDESC+4                   ; 0559
                               08   AC  DD 00021          PUSHL   BINTIM                                   ; 0560
                               1C   AE  9F 00024          PUSHAB  TIMEDESC                                 ;
                  00000000G  00     02  FB 00027          CALLS   #2, SYS$BINTIM                           ;
                             50     01  D0 0002E          MOVL    #1, R0                                   ; 0561
                                        04 00031          RET                                              ; 0562

; Routine Size:  50 bytes,    Routine Base:  $CODE$ + 0311
;  452       0563  1 global routine lnk$addimage (moduledesc, modulerfa, retcludesc, foundflag) =
;  453       0564  2     begin
;  454       0565  2 !
;  455       0566  2 !       THIS ROUTINE IS CALLED BY THE LIBRARIAN WHEN IT FINDS A MODULE
;  456       0567  2 !       NAME WITH THE SAME RFA AS THE GLOBAL SYMBOL JUST LOCATED.  WE
;  457       0568  2 !       CHECK TO SEE IF THIS SHAREABLE IMAGE HAS ALREADY BEEN REQUESTED.
;  458       0569  2 !       IF NOT, THEN A CLUSTER DESCRIPTOR AND FDB ARE ALLOCATED.
;  459       0570  2 !
;  460       0571  2 !       IF MODULERFA IS NOT PRESENT (NULLPARAMETER), THEN NO LIBRARY READING
;  461       0572  2 !       IS DONE, THE CLUSTER DESCRIPTOR AND FILE DESCRIPTOR BLOCKS ARE CREATED,
;  462       0573  2 !       HOWEVER.
;  463       0574  2 !
;  464       0575  2 !       IF RETCLUDESC IS PASSED, IT IS THE ADDRESS OF A LONGWORD TO STORE
;  465       0576  2 !       THE ALLOCATED CLUSTER DESCRIPTOR ADDRESS.  NOTE THAT THE ONLY WAY
;  466       0577  2 !       TO DETERMINE IF AN IMAGE WAS REQUESTED IS TO CHECK FOR RETCLUDESC
;  467       0578  2 !       BEING NON-0, SINCE THIS ROUTINE ALWAYS RETURNS FALSE TO STOP LBR SEARCH
;  468       0579  2 !
;  469       0580  2 !       IF FOUNDFLAG IS PASSED, IT IS THE ADDRESS OF A LONGWORD TO STORE
;  470       0581  2 !       A 1 (FOUND) OR 0 (INSERTED)
;  471       0582  2 !
;  472       0583  2     routine compareclu (keydesc, clunode) =
;  473       0584  3         begin
;  474       0585  3 !
;  475       0586  3 ! LOCAL ROUTINE TO COMPARE A NAME OF NODE WITH ANOTHER NAME
;  476       0587  3 !
;  477       0588  3         map
;  478       0589  3             keydesc : ref block [, byte],        ! POINTER TO STRING DESCRIPTOR
;  479       0590  3             clunode : ref block [, byte];        ! NODE FOR DESCRIPTOR BEING EXAMINED
;  480       0591  3         local
;  481       0592  3             clu : ref block [, byte];
;  482       0593  3         clu = .clunode [node$l_ptr];             ! POINT TO CLUSTER DESCRIPTOR
;  483       0594  3         return ch$compare (.keydesc [dsc$w_length], .keydesc [dsc$a_pointer], .clu [clu$b_namlng],
;  484       0595  3                 clu [clu$t_name])
;  485       0596  2         end;



                                               001C 00000 COMPARECLU:
                                                                  .WORD   Save R2,R3,R4                              ; 0583
                         50      08  AC  D0 00002                 MOVL    CLUNODE, R0                                ; 0593
                         50      0A  A0  D0 00006                 MOVL    10(R0), CLU                                ;
                         51      04  AC  D0 0000A                 MOVL    KEYDESC, R1                                ; 0594
                         52      5C  A0  9A 0000E                 MOVZBL  92(CLU), R2                                ;
                         54          01  D0 00012                 MOVL    #1, R4                                     ; 0595
        52       00      04      B1      61  2D 00015             CMPC5   (R1), @4(R1), #0, R2, 93(CLU)              ;
                                 5D  A0     0001B                                                                    ;
                                     03  1A 0001D                 BGTRU   1$                                         ;
                         54          01  D9 0001F                 SBWC    #1, R4                                     ;
                         50          54  D0 00022 1$:             MOVL    R4, R0                                     ;
                                         04 00025                 RET                                                ; 0596

; Routine Size:  38 bytes,    Routine Base:  $CODE$ + 0343


;  486       0597  2 !
```

```
2 ! MAIN BODY OF LNK$ADDIMAGE
2 !
2     map
2         moduledesc : ref block [, byte],
2         modulerfa : ref block [, byte],
2         retcludesc : ref vector [, long],
2         foundflag : ref vector [, long];
2     builtin
2         nullparameter;
2     local
2         status,
2         read_library,
2         mhdbuf : block [lbr$c_maxhdrsiz, byte], ! BUFFER TO READ MODULE HEADER
2         mhdbufdesc : block [dsc$c_s_bln, byte], ! STRING DESCRIPTOR OF MHDBUF
2         bufdesc : block [dsc$c_s_bln, byte],
2         prevclu : ref block [, byte],         ! POINTER TO PREVIOUS CLUSTER
2         nextclu : ref block [, byte],
2         lastclu : ref block [, byte],
2         fdb : ref block [, byte],             ! POINTER TO CREATED FDB
2         clu : ref block [, byte];             !  AND CLUSTER
2
2 !
2 ! SEARCH THE CLUSTER LIST TO SEE IF THIS SHAREABLE IMAGE ALREADY REQUESTED
2 !
2     if not nullparameter (3) then retcludesc [0] = 0;
2     if not nullparameter (4) then foundflag [0] = 0;
2     if lib$lookup_tree (lnk$gl_clutree, .moduledesc, compareclu, clu)   ! LOOK IT UP
2     then
3         begin
3         if not nullparameter (4) then foundflag [0] = 1;
3         if not nullparameter (3) then retcludesc [0] = .clu [node$l_ptr];
3         return lnk$k_stopsearch;                          !  RETURN FALSE TO STOP SEARCH IF FOUND
2         end;
2 !
2 ! IMAGE NOT REQUESTED.  READ AND VERIFY MODULE HEADER.  THEN CREATE CLUSTER DESCRIPTOR
2 !
3     if (read_library = not nullparameter (2))   ! SET FLAG IF PARAMETER SPECIFIED
2     then
3         begin
3
4         if not (status = lbr$find (%ref (.lnk$gl_curfil [fdb$w_ifi]), .modulerfa))      ! POINT TO THE MODUL
3         then
4             begin
4             signal ((lin$_libfind and not sts$m_severity) or sts$k_error, 4,    ! REPORT ERROR
4                 .modulerfa [rfa$l_vbn], .modulerfa [rfa$w_offset], .moduledesc [dsc$a_pointer] - 1,
4                 lnk$gl_curfil [fdb$q_filename], lin$_format, 0, .status, .lbr$gl_rmsstv);
4             return lnk$k_stopsearch;                      ! RETURN TO STOP SEARCH
3             end;
3
3         bufdesc [dsc$w_length] = .lnk$al_rab [rab$w_usz];
3                                                   ! SET UP BUFFER DESCRIPTOR TO READ OBJ MODULE HEADER REC
3         bufdesc [dsc$a_pointer] = .lnk$al_rab [rab$l_ubf];
4         if not (status = lbr$get_record (%ref (.lnk$gl_curfil [fdb$w_ifi]), bufdesc, bufdesc))
4                                                   ! READ FIRST RECORD OF OBJ MODULE
3         then
4             begin
4             signal (lin$_readerr, 1, lnk$gl_curfil [fdb$q_filename], .status, .lbr$gl_rmsstv);
```

```
4              return lnk$k_stopsearch;
3              end;
3          mhdbufdesc [dsc$w_length] = lbr$c_maxhdrsiz;
3                                                       ! READ LIBRARY MODULE HEADER...SET UP BUFFER DESCRIPTOR
3          mhdbufdesc [dsc$a_pointer] = mhdbuf;
4          if not (status = lbr$set_module (%ref (.lnk$gl_curfil [fdb$w_ifi]), .modulerfa,          ! READ IT NO
4                  mhdbufdesc, mhdbufdesc))
3          then
4              begin
4              signal (lin$_readerr, 1, lnk$gl_curfil [fdb$q_filename], .status, .lbr$gl_rmsstv);
4              return lnk$k_stopsearch;
3              end;
4          begin
4          bind
4              hdrec = .bufdesc [dsc$a_pointer] : block [, byte],  ! NAME THE HEADER RECORD
4              mhdid = hdrec [mhd$t_name] + .hdrec [mhd$b_namlng] : vector [, byte];
4                                                       ! AND THE MODULE ID PART OF HEADER
4          if .hdrec [obj$b_rectyp] neq obj$c_hdr  ! MAKE SURE IT LOOKS LIKE AN OBJ MODULE HEADER
4              or .hdrec [obj$b_subtyp] neq obj$c_hdr_mhd
4          then
5              begin
5              signal (lin$_readerr, 1, lnk$gl_curfil [fdb$q_filename], lin$_format, 0);
5              return lnk$k_stopsearch;
4              end;
4
3          end;
2          end;                                         ! OF READ_LIBRARY
2  !
2  ! NOW ALLOCATE A CLUSTER DESCRIPTOR FOR THE NEW SHAREABLE IMAGE
2  !
2      lnk$allocluster (clu, 1);                        ! CREATE CLUSTER DESCRIPTOR, DON'T LINK INTO LIST
2      if not nullparameter (3)                         ! IF CALLER WANTS DESCRIPTOR ADDRESS
2      then
2          retcludesc [0] = .clu;                       !  THEN RETURN IT
2
2      lastclu = .lnk$gl_curclu [clu$l_lastclu];        ! GET POINTER TO LAST IMAGE CONTAINED IN THIS ONE
2
2      if .lastclu neq 0                                ! IF THERE IS ONE, INSERT AFTER IT
2      then
3          begin
3          nextclu = .lastclu [clu$l_nxtclu];
3          lastclu [clu$l_nxtclu] = .clu;
3          clu [clu$l_prevclu] = .lastclu;
3          end
2      else
3          begin                                        ! THIS IS THE FIRST, INSERT AFTER CURRENT CLUSTER
3          nextclu = .lnk$gl_curclu [clu$l_nxtclu];
3          lnk$gl_curclu [clu$l_nxtclu] = .clu;
3          clu [clu$l_prevclu] = .lnk$gl_curclu;
2          end;
2
2      if (clu [clu$l_nxtclu] = .nextclu) neq 0         ! SET PREVCLU IN NEXT CLUSTER
2      then
2          nextclu [clu$l_prevclu] = .clu
2      else
2          lnk$gl_lastclu = .clu;                       ! OR MAKE THIS THE LAST CLUSTER IF IT IS
2
```

```
2         lnk$gl_curclu [clu$l_lastclu] = .clu;          ! MAKE THIS THE NEW LAST CLUSTER
2         clu [clu$v_shrimg] = true;                     ! FLAG CLUSTER AS SHAREABLE IMAGE
2         clu [clu$v_intclu] = true;                     ! FLAG AS INTERNALLY CREATED
2         ch$move ((clu [clu$b_namlng] = .moduledesc [dsc$w_length]),
2                                                        ! SET MODULE NAME INTO CLUSTER DESCRIPTOR
2             .moduledesc [dsc$a_pointer], clu [clu$t_name]);
2         lnk$insert_clu (.clu);                         ! INSERT CLUSTER INTO CLUSTER TREE
2
2         if .read_library                               ! IF READING LIBRARY, SET MORE INFO INTO CLUSTER DESCRIPTOR
2         then
3             begin
3             bind
3                 hdrec = .bufdesc [dsc$a_pointer] : block [, byte],   ! NAME THE HEADER RECORD
3                 mhdid = hdrec [mhd$t_name] + .hdrec [mhd$b_namlng] : vector [, byte],
3                                                        ! AND THE MODULE ID PART OF HEADER
3                 mhdgsmatch = mhdid [1] : long,         !
3                 mhdcredat = mhdid [0] + .mhdid [0] + 1 : vector [, byte],   ! AND THE CREATE DATE/TIME
3                 modheader = .mhdbufdesc [dsc$a_pointer] : block [, byte],   ! THE LIBRARY MODULE HEADER
3                 modgsmatch = modheader [mhd$t_objident] : long;      ! THE GSMATCH STORED IN LIBRARY MODULE HEADE
3
3             lnk$bintim (mhdcredat, clu [clu$q_credat]);      ! CONVERT CREATION DATE/TIME FOR LATER
3             clu [clu$l_gsmatch] = .modgsmatch;       ! SAVE THE GSMATCH FOUND IN THE LIBRARY
3             end;
2     !
2     ! ALLOCATE AN FDB
2     !
2         lnk$allofdb (fdb);
2         clu [clu$l_fstfdb] = clu [clu$l_lstfdb] = .fdb;
2         lnk$alloblk ((fdb [fdb$w_usrnamlen] = .moduledesc [dsc$w_length]), fdb [fdb$l_usrnamadr]);
2         ch$move (.fdb [fdb$w_usrnamlen], .moduledesc [dsc$a_pointer], .fdb [fdb$l_usrnamadr]);
2
2         if .lnk$gl_ctlmsk [lnk$v_intfil]
2         then
2             ch$move (dsc$c_s_bln, shrdefext, fdb [fdb$w_defnamlen]) ! SET DEFAULT FILENAME STRING
2         else
3             begin
3             local
3                 ptr,
3                 ptr1;
3     !
3     ! THE DEFAULT FILENAME STRING CONSISTS OF THE RESULTANT
3     ! FILENAME OF THE CURRENT FILE WITH THE EXTENSION SET TO ".EXE"
3     !
4             if (ptr = ch$find_ch (.lnk$gl_curfil [fdb$w_defnamlen],         ! FIND END OF DIRECTORY
3                     .lnk$gl_curfil [fdb$l_defnamadr], %ascii']')) eql 0
3             then
3                 ptr = ch$find_ch (.lnk$gl_curfil [fdb$w_defnamlen], .lnk$gl_curfil [fdb$l_defnamadr], %ascii'>')
3
3             ptr1 = ch$find_ch (.lnk$gl_curfil [fdb$w_defnamlen] - (.ptr - .lnk$gl_curfil [fdb$l_defnamadr]),
3                 .ptr, %ascii'.');                      ! THEN FIND START OF EXTENSION
3             lnk$alloblk (.ptr1 - .lnk$gl_curfil [fdb$l_defnamadr] + 4, fdb [fdb$l_defnamadr]);
3                                                        ! ALLOCATE BLOCK TO HOLD MODIFIED NAME(+4 FOR .EXE)
3             ptr = ch$move (.ptr1 - .lnk$gl_curfil [fdb$l_defnamadr], .lnk$gl_curfil [fdb$l_defnamadr],
3                                                        ! MOVE IN FIRST PART OF NAME
3                 .fdb [fdb$l_defnamadr]);
3             ptr1 = ch$move (4, uplit ('.EXE'), .ptr);         ! SET THE EXTENSION
3             fdb [fdb$w_defnamlen] = .ptr1 - .fdb [fdb$l_defnamadr]; ! COMPUTE LENGTH OF DEFAULT NAME
```

```
;  658        0769  2         end;
;  659        0770  2     ch$move (dsc$c_s_bln, lnk$gl_curfil [fdb$q_filename], fdb [fdb$q_libnamdsc]);
;  660        0771  2                                                     ! COPY LIBRARY FILE DESCRIPTOR
;  661        0772  2     fdb [fdb$v_shr] = true;                         ! FLAG FILE AS SHAREABLE IMAGE
;  662        0773  2     return lnk$k_stopsearch                         ! RETURN FALSE TO STOP SEARCH
;  663        0774  1     end;                                            ! OF ADDIMAGE


                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

                         45  58  45  2E  00010 P.AAB:   .ASCII  \.EXE\                                       ;


                                                        .PSECT  $CODE$,NOWRT,2

                                    0FFC 00000          .ENTRY  LNK$ADDIMAGE, Save R2,R3,R4,R5,R6,R7,R8,R9,-; 0563
                                                                R10,R11                                      :
                      5B 00000000G  8F  D0 00002        MOVL    #LIN$_FORMAT, R11                            :
                      5A 00000000G  00  9E 00009        MOVAB   LBR$GL_RMSSTV, R10                           :
                      59 00000000G  00  9E 00010        MOVAB   LNK$GL_CURFIL, R9                            :
                      5E      FF64  CE  9E 00017        MOVAB   -156(SP), SP                                 :
                      03            6C  91 0001C        CMPB    (AP), #3                                     : 0622
                                    08  1F 0001F        BLSSU   1$                                           :
                              0C    AC  D5 00021        TSTL    12(AP)                                       :
                                    03  13 00024        BEQL    1$                                           :
                              0C    BC  D4 00026        CLRL    @RETCLUDESC                                  :
                      04            6C  91 00029 1$:    CMPB    (AP), #4                                     : 0623
                                    08  1F 0002C        BLSSU   2$                                           :
                              10    AC  D5 0002E        TSTL    16(AP)                                       :
                                    03  13 00031        BEQL    2$                                           :
                              10    BC  D4 00033        CLRL    @FOUNDFLAG                                   :
                              04    AE  9F 00036 2$:    PUSHAB  CLU                                          : 0624
                              9E    AF  9F 00039        PUSHAB  COMPARECLU                                   :
                      58      04    AC  D0 0003C        MOVL    MODULEDESC, R8                               :
                                    58  DD 00040        PUSHL   R8                                           :
                         00000000G  00  9F 00042        PUSHAB  LNK$GL_CLUTREE                               :
             00000000G 00           04  FB 00048        CALLS   #4, LIB$LOOKUP_TREE                          :
                       23           50  E9 0004F        BLBC    R0, 4$                                       :
                       04           6C  91 00052        CMPB    (AP), #4                                     : 0627
                                    09  1F 00055        BLSSU   3$                                           :
                              10    AC  D5 00057        TSTL    16(AP)                                       :
                                    04  13 0005A        BEQL    3$                                           :
                  10   BC           01  D0 0005C        MOVL    #1, @FOUNDFLAG                               :
                       03           6C  91 00060 3$:    CMPB    (AP), #3                                     : 0628
                                    70  1F 00063        BLSSU   8$                                           :
                              0C    AC  D5 00065        TSTL    12(AP)                                       :
                                    6B  13 00068        BEQL    8$                                           :
                       50     04    AE  D0 0006A        MOVL    CLU, R0                                      :
                  0C   BC     0A    A0  D0 0006E        MOVL    10(R0), @RETCLUDESC                          :
                                    60  11 00073        BRB     8$                                           : 0629
                       02           6C  91 00075 4$:    CMPB    (AP), #2                                     : 0634
                                    05  1E 00078        BGEQU   5$                                           :
                       56           01  D0 0007A        MOVL    #1, R6                                       :
                                    09  11 0007D        BRB     6$                                           :
                                    56  D4 0007F 5$:    CLRL    R6                                           :
```

```
LNK_PROCSLIB                                                  I  7
V04-000                                               16-Sep-1984 00:21:45    VAX-11 Bliss-32 V4.0-742              Page  20
                                                      14-Sep-1984 12:40:34    [LINKER.SRC]LNKPROLIB.B32;1                (4)

                                     08    AC  D5 00081           TSTL    8(AP)                             :
                                           02  12 00084           BNEQ    6$                                :
                                           56  D6 00086           INCL    R6                                :
                        56                 56  D2 00088  6$:      MCOML   R6, READ_LIBRARY                  :
                        03                 56  E8 0008B           BLBS    READ_LIBRARY, 7$                  :
                                         00CC  31 0008E           BRW     14$                               :
                        52           08    AC  D0 00091  7$:      MOVL    MODULERFA, R2                     : 0638
                                           52  DD 00095           PUSHL   R2                                :
                        50                 69  D0 00097           MOVL    LNK$GL_CURFIL, R0                 :
              04        AE           24    A0  3C 0009A           MOVZWL  36(R0), 4(SP)                     :
                                     04    AE  9F 0009F           PUSHAB  4(SP)                             :
       00000000G        00                 02  FB 000A2           CALLS   #2, LBR$FIND                      :
                        53                 50  D0 000A9           MOVL    R0, STATUS                        :
                        29                 53  E8 000AC           BLBS    STATUS, 9$                        :
                                           6A  DD 000AF           PUSHL   LBR$GL_RMSSTV                     : 0643
                                           53  DD 000B1           PUSHL   STATUS                            :
                                           7E  D4 000B3           CLRL    -(SP)                             :
                                           5B  DD 000B5           PUSHL   R11                               :
   7E                   69                 14  C1 000B7           ADDL3   #20, LNK$GL_CURFIL, -(SP)         :
   7E         04        A8                 01  C3 000BB           SUBL3   #1, 4(R8), -(SP)                  : 0642
                        7E           04    A2  3C 000C0           MOVZWL  4(R2), -(SP)                      : 0643
                                           62  DD 000C4           PUSHL   (R2)                              :
                                           04  DD 000C6           PUSHL   #4                                :
                            00000000*      8F  DD 000C8           PUSHL   #<<LIN$_LIBFIND&-8>!2>            :
       00000000G        00                 0A  FB 000CE           CALLS   #10, LIB$SIGNAL                   :
                                         01DA  31 000D5  8$:      BRW     27$                               : 0644
              0C        AE  00000000G      00  B0 000D8  9$:      MOVW    LNK$AL_RAB+32, BUFDESC            : 0647
              10        AE  00000000G      00  D0 000E0           MOVL    LNK$AL_RAB+36, BUFDESC+4          : 0649
                                     0C    AE  9F 000E8           PUSHAB  BUFDESC                           : 0650
                                     10    AE  9F 000EB           PUSHAB  BUFDESC                           :
                        50                 69  D0 000EE           MOVL    LNK$GL_CURFIL, R0                 :
              08        AE           24    A0  3C 000F1           MOVZWL  36(R0), 8(SP)                     :
                                     08    AE  9F 000F6           PUSHAB  8(SP)                             :
       00000000G        00                 03  FB 000F9           CALLS   #3, LBR$GET_RECORD                :
                        53                 50  D0 00100           MOVL    R0, STATUS                        :
                        2A                 53  E9 00103           BLBC    STATUS, 10$                       :
              14        AE           80    8F  9B 00106           MOVZBW  #128, MHDBUFDESC                  : 0657
              18        AE           1C    AE  9E 0010B           MOVAB   MHDBUF, MHDBUFDESC+4              : 0659
                                     14    AE  9F 00110           PUSHAB  MHDBUFDESC                        : 0660
                                     18    AE  9F 00113           PUSHAB  MHDBUFDESC                        :
                                           52  DD 00116           PUSHL   R2                                :
                        50                 69  D0 00118           MOVL    LNK$GL_CURFIL, R0                 :
              0C        AE           24    A0  3C 0011B           MOVZWL  36(R0), 12(SP)                    :
                                     0C    AE  9F 00120           PUSHAB  12(SP)                            :
       00000000G        00                 04  FB 00123           CALLS   #4, LBR$SET_MODULE                :
                        53                 50  D0 0012A           MOVL    R0, STATUS                        :
                        06                 53  E8 0012D           BLBS    STATUS, 11$                       :
                                           6A  DD 00130  10$:     PUSHL   LBR$GL_RMSSTV                     : 0664
                                           53  DD 00132           PUSHL   STATUS                            :
                                           11  11 00134           BRB     13$                               :
                        50           10    AE  D0 00136  11$:     MOVL    BUFDESC+4, R0                     : 0669
                                           60  95 0013A           TSTB    (R0)                              : 0672
                                           05  12 0013C           BNEQ    12$                               :
                                     01    A0  95 0013E           TSTB    1(R0)                             : 0673
                                           1A  13 00141           BEQL    14$                               :
                                           7E  D4 00143  12$:     CLRL    -(SP)                             : 0676
                                           5B  DD 00145           PUSHL   R11                               :
```

```
LNK_PROCSLIB                                           J  7
V04-000                                      16-Sep-1984 00:21:45    VAX-11 Bliss-32 V4.0-742            Page  21
                                             14-Sep-1984 12:40:34    [LINKER.SRC]LNKPROLIB.B32;1               (4)

      7E              69               14  C1 00147 13$:   ADDL3   #20, LNK$GL_CURFIL, -(SP)          ;
                                       01  DD 0014B        PUSHL   #1                                 ;
                           00000000G   8F  DD 0014D        PUSHL   #LIN$_READERR                      ;
            00000000G 00               05  FB 00153        CALLS   #5, LIB$SIGNAL                     ;
                                     0155  31 0015A        BRW     27$                                ; 0677
                                       01  DD 0015D 14$:   PUSHL   #1                                 ; 0685
                                  08   AE  9F 0015F        PUSHAB  CLU                                ;
            00000000G 00               02  FB 00162        CALLS   #2, LNK$ALLOCLUSTER                ;
                      03               6C  91 00169        CMPB    (AP), #3                           ; 0686
                                       0A  1F 0016C        BLSSU   15$                                ;
                                  0C   AC  D5 0016E        TSTL    12(AP)                             ;
                                       05  13 00171        BEQL    15$                                ;
                   0C BC          04   AE  D0 00173        MOVL    CLU, @RETCLUDESC                   ; 0688
                      50   00000000G   00  D0 00178 15$:   MOVL    LNK$GL_CURCLU, R0                  ; 0690
                      52          24   A0  D0 0017F        MOVL    36(R0), LASTCLU                    ;
                      57          04   AE  D0 00183        MOVL    CLU, R7                            ; 0696
                                       52  D5 00187        TSTL    LASTCLU                            ; 0692
                                       0C  13 00189        BEQL    16$                                ;
                      51               62  D0 0018B        MOVL    (LASTCLU), NEXTCLU                 ; 0695
                      62               57  D0 0018E        MOVL    R7, (LASTCLU)                      ; 0696
                   04 A7               52  D0 00191        MOVL    LASTCLU, 4(R7)                     ; 0697
                                       0A  11 00195        BRB     17$                                ; 0692
                      51               60  D0 00197 16$:   MOVL    (R0), NEXTCLU                      ; 0701
                      60               57  D0 0019A        MOVL    R7, (R0)                           ; 0702
                   04 A7               50  D0 0019D        MOVL    R0, 4(R7)                          ; 0703
                      67               51  D0 001A1 17$:   MOVL    NEXTCLU, (R7)                      ; 0706
                                       06  13 001A4        BEQL    18$                                ;
                   04 A1               57  D0 001A6        MOVL    R7, 4(NEXTCLU)                     ; 0708
                                       07  11 001AA        BRB     19$                                ;
            00000000G 00               57  D0 001AC 18$:   MOVL    R7, LNK$GL_LASTCLU                 ; 0710
                   24 A0               57  D0 001B3 19$:   MOVL    R7, 36(R0)                         ; 0712
                   58 A7        0204   8F  A8 001B7        BISW2   #516, 88(R7)                       ; 0714
                      50               68  3C 001BD        MOVZWL  (R8), R0                           ; 0715
                   5C A7               50  90 001C0        MOVB    R0, 92(R7)                         ;
5D    A7           04 B8               50  28 001C4        MOVC3   R0, @4(R8), 93(R7)                 ; 0717
                                       57  DD 001CA        PUSHL   R7                                 ; 0718
            00000000G 00               01  FB 001CC        CALLS   #1, LNK$INSERT_CLU                 ;
                      26               56  E9 001D3        BLBC    READ_LIBRARY, 20$                  ; 0720
                      51          10   AE  D0 001D6        MOVL    BUFDESC+4, R1                      ; 0724
                      50          05   A1  9A 001DA        MOVZBL  5(R1), R0                          ; 0725
                      50     06 A140       9E 001DE        MOVAB   6(R1)[R0], R0                      ;
                      51               60  9A 001E3        MOVZBL  (R0), R1                           ; 0728
      52           18 AE               12  C1 001E6        ADDL3   #18, MHDBUFDESC+4, R2              ; 0730
                                  30   A7  9F 001EB        PUSHAB  48(R7)                             ; 0732
                             01 A140       9F 001EE        PUSHAB  1(R1)[R0]                          ;
                 FDB1 CF               02  FB 001F2        CALLS   #2, LNK$BINTIM                     ;
                 00B4 C7               62  D0 001F7        MOVL    (R2), 132(R7)                      ; 0733
                                  08   AE  9F 001FC 20$:   PUSHAB  FDB                                ; 0738
            00000000G 00               01  FB 001FF        CALLS   #1, LNK$ALLOFDB                    ;
                      56          08   AE  D0 00206        MOVL    FDB, R6                            ; 0739
                   0C A7               56  D0 0020A        MOVL    R6, 12(R7)                         ;
                   08 A7               56  D0 0020E        MOVL    R6, 8(R7)                          ;
                                  10   A6  9F 00212        PUSHAB  16(R6)                             ; 0740
                      7E               68  3C 00215        MOVZWL  (R8), -(SP)                        ;
                   0C A6               6E  B0 00218        MOVW    (SP), 12(R6)                       ;
            00000000G 00               02  FB 0021C        CALLS   #2, LNK$ALLOBLK                    ;
10    B6           04 B8          0C   A6  28 00223        MOVC3   12(R6), @4(R8), @16(R6)            ; 0741
```

```
           0B 00000000G   00           03  E1 0022A          BBC      #3, LNK$GL_CTLMSK+1, 21$              ; 0743
    14     A6 00000000'   EF           08  28 00232          MOVC3    #8, SHRDEFEXT, 20(R6)                 ; 0745
                                       68  11 0023B          BRB      26$                                   ;
                          52           69  D0 0023D 21$:     MOVL     LNK$GL_CURFIL, R2                     ; 0755
    18     B2       14    A2     5D    8F  3A 00240          LOCC     #93, 20(R2), @24(R2)                  ;
                                       02  12 00247          BNEQ     22$                                   ;
                                       51  D4 00249          CLRL     R1                                    ;
                          53           51  D0 0024B 22$:     MOVL     R1, PTR                               ;
                                       0D  12 0024E          BNEQ     24$                                   ; 0756
    18     B2       14    A2           3E  3A 00250          LOCC     #62, 20(R2), @24(R2)                  ; 0758
                                       02  12 00256          BNEQ     23$                                   ;
                                       51  D4 00258          CLRL     R1                                    ;
                          53           51  D0 0025A 23$:     MOVL     R1, PTR                               ;
           50       18    A2           53  C3 0025D 24$:     SUBL3    PTR, 24(R2), R0                       ; 0760
                          51     14    A2  3C 00262          MOVZWL   20(R2), R1                            ;
                          50           51  C0 00266          ADDL2    R1, R0                                ;
           63             50           2E  3A 00269          LOCC     #46, R0, (PTR)                        ;
                                       02  12 0026D          BNEQ     25$                                   ;
                                       51  D4 0026F          CLRL     R1                                    ;
                          57           51  D0 00271 25$:     MOVL     R1, PTR1                              ;
                                 18    A6  9F 00274          PUSHAB   24(R6)                                ; 0762
           51             57     18    A2  C3 00277          SUBL3    24(R2), PTR1, R1                      ;
                                 04    A1  9F 0027C          PUSHAB   4(R1)                                 ;
              00000000G   00           02  FB 0027F          CALLS    #2, LNK$ALLOBLK                       ;
                          50           69  D0 00286          MOVL     LNK$GL_CURFIL, R0                     ; 0764
           51             57     18    A0  C3 00289          SUBL3    24(R0), PTR1, R1                      ;
    18     B6       18    B0           51  28 0028E          MOVC3    R1, @24(R0), @24(R6)                  ; 0766
                          63 00000000' EF  D0 00294          MOVL     P.AAB, (PTR)                          ; 0767
                          57     04    A3  9E 0029B          MOVAB    4(R3), PTR1                           ;
    14     A6             57     18    A6  A3 0029F          SUBW3    24(R6), PTR1, 20(R6)                  ; 0768
                          50           69  D0 002A5 26$:     MOVL     LNK$GL_CURFIL, R0                     ; 0770
    1C     A6       14    A0           08  28 002A8          MOVC3    #8, 20(R0), 28(R6)                    ;
                    0A    A6           04  88 002AE          BISB2    #4, 10(R6)                            ; 0772
                                       50  D4 002B2 27$:     CLRL     R0                                    ; 0774
                                       04     002B4          RET                                            ;

; Routine Size:  693 bytes,    Routine Base:  $CODE$ + 0369


;  664          0775  1
;  665          0776  1 end
;  666          0777  0 eludom



                                                                     .EXTRN  LIB$SIGNAL

;                                   PSECT SUMMARY
;
;       Name                     Bytes                       Attributes
;
;  $PLIT$                           20  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $OWN$                            24  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $GLOBAL$                         20  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                         1566  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  .  ABS  .                         0  NOVEC,NOWRT,NORD ,NOEXE,NOSHR,  LCL,  ABS,  CON,NOPIC,ALIGN(0)
;                                  Library Statistics
;
;                                                 -------- Symbols --------          Pages        Processing
;       File                                      Total     Loaded    Percent        Mapped       Time
;
;  _$255$DUA28:[SYSLIB]STARLET.L32;1              9776         24          0           581          00:01.0
;  _$255$DUA28:[LINKER.OBJ]DATBAS.L32;1            538         38          7            28          00:00.5



;                                          COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:LNKPROLIB/OBJ=OBJ$:LNKPROLIB MSRC$:LNKPROLIB/UPDATE=(ENH$:LNKPROLIB)

;  667          0778  0                                                          ! End of module
; Size:         1566 code + 64 data bytes
; Run Time:        00:28.7
; Elapsed Time:    01:01.8
; Lines/CPU Min:    1624
; Lexemes/CPU-Min: 17402
; Memory Used:  242 pages
; Compilation Complete
```

0219 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

LNKPROLIB
LIS

LNKSYMTBL
LIS

LNKSYMOUT
LIS

LNKVMALLO
LIS

LNKPSCTBL
LIS

LNKPROSHR
LIS

LNKSTATSO
LIS